Faster and Less Error-Prone: Supplementing an Accessible Keyboard with Speech Input

Bhakti Bhikne, Anirudha Joshi, Manjiri Joshi, Charudatta Jadhav, Prabodh Sakhardande

▶ **To cite this version:**

Bhakti Bhikne, Anirudha Joshi, Manjiri Joshi, Charudatta Jadhav, Prabodh Sakhardande. Faster and Less Error-Prone: Supplementing an Accessible Keyboard with Speech Input. 17th IFIP Conference on Human-Computer Interaction (INTERACT), Sep 2019, Paphos, Cyprus. pp.288-304, 10.1007/978-3-030-29381-9_18 . hal-02544545

HAL Id: hal-02544545
https://inria.hal.science/hal-02544545

Submitted on 16 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Faster and Less Error-prone: Supplementing an Accessible Keyboard with Speech Input

Bhakti Bhikne[1], Anirudha Joshi[1], Manjiri Joshi[1], Dr. Charudatta Jadhav[2], Prabodh Sakhardande[1]

[1] Industrial Design Centre, IIT Bombay, Mumbai, India
[2] Tata Consultancy Services Limited, Mumbai, Maharashtra, India
mailbhaktibhikne@gmail.com, anirudha@iitb.ac.in, manjirij@iitb.ac.in, charudatta.jadhav@tcs.com, prabodh.sakhardande@gmail.com

Abstract. Swarachakra is an Abugida text input keyboard available in 12 Indian languages. We enhanced an accessible version of Swarachakra Marathi with speech input. However, speech input could be error-prone, and especially so for languages where speech recognition technologies are new. Such errors could either slow the user down due to the need for editing, or go unnoticed, leading to high uncorrected error rates. We therefore conducted a within-subject empirical study to compare the user performance of keyboard-only input method with keyboard+speech input method with 11 novice visually impaired users. We found that keyboard+speech input was almost 11 times faster, reaching 182 characters per minute, and had a lower uncorrected error rate than the keyboard-only input, and in spite of having higher corrected error rates. Though we used a wide variety of phrases in our study, we observed that all phrases were faster on average with the keyboard+speech input method. To the best of our knowledge, ours is the first empirical study to evaluate the performance of speech enabled text input in Marathi for visually impaired people. This is the highest reported speed by visually impaired users in any Indian language.

Keywords: Speech-based text entry, Accessibility, Longitudinal Study, Visually Impaired users.

1 Introduction

In this paper, we investigate the question, "With the advancement of speech recognition technologies, can speech augment text input by visually impaired users in Indian languages?"

Despite technological advancements, text input for the visually impaired people remains a hurdle. Although there has been a widespread adoption of smartphones and screen-readers such as Talkback [11] and VoiceOver [12] by visually impaired users, typing on mobile phones remains slow and laborious for them [1, 2, 3, 7]. This is even more so for Indian languages. In recent years, research for text input in Indian languages by sighted users has gathered steam [5, 6, 7, 8, 10, 13, 14, and 15]. On the other hand, research in text input in Indian languages by visually impaired people is notably under-developed.

Text entry in Indian languages has always been a challenge for users, including low speeds and high error rates, mainly due to the complex structure of the Devanagari script. Studies with sighted users were reported to have text input speeds between 35 to 45 characters per minute (CPM) on four keyboards [6]. The only study for text input by visually impaired users for an Indian language reported 15 CPM using the Swarachakra keyboard and 13 CPM using the Google Indic keyboard [7]. In our recent study, we found that enhancing the keyboard with speech input could enable sighted users achieve mean speeds of 118 CPM in Hindi [8]. In this paper, we investigate if we could we achieve similar improvements in performance for visually impaired users. We found that enhancing a keyboard with speech increases the text input speed of visually impaired users by about 11 times compared to keyboard-only input, reaching a mean of 182 CPM (figure 1). This is the highest reported speed by visually impaired users in any Indian language.

In the next section, we discuss the background related to our work. Next, we introduce the keyboards and the method we used for our study. We next present our results, and finally present our conclusions.

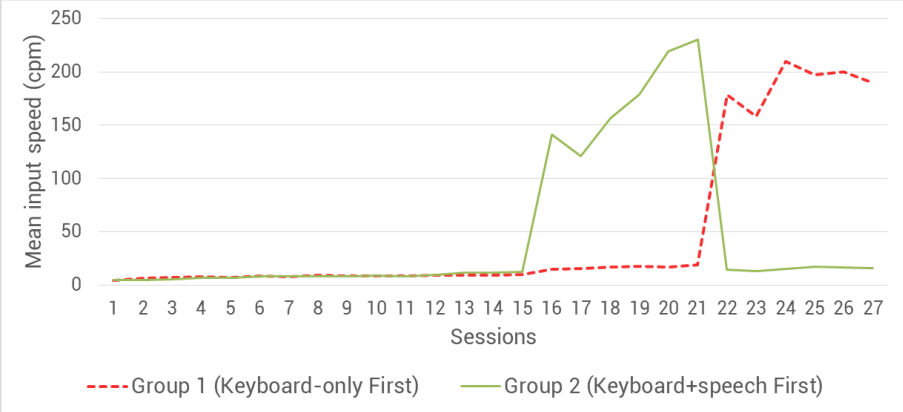

Fig. Error! No sequence specified.**.** The improvement in text entry rates due to speech as seen by the large peaks when speech is used as an input method between sessions 16 to 21 for group 1 and between sessions 22 to 27 for group 2.

2 Background

Our work deals with Marathi, a language spoken in India by about 72 million native speakers and 84 million total speakers [19]. Marathi uses the Devanagari script, which is an Abugida script used by several other languages including Hindi, Konkani,

Kashmiri, and Sanskrit [20]. Users have found Devanagari, and other Abugidas scripts challenging to input on digital devices, something that has been extensively discussed in literature [5, 6, and 7]. We summarise below the key challenges for the purpose of completeness.

Typically, there are four types of glyphs in the Devanagari script, which can combine together to form words. Firstly, a consonant in Devanagari may stand alone as an independent glyph, with an inherent vowel. Likewise, a vowel can stand independently. Thirdly, a consonant (C) may be "modified" by a vowel modifier (V), leading to a C+V glyph. In such a glyph, the vowel modifier may appear before, after, above or below the consonant, which often causes confusions in the mind of the users about the sequence they should use to input the vowel modifier. Fourthly, two or more consonants can combine to form a conjunct, which in turn may have a vowel modifier. Conjuncts are particularly difficult for users, as, at times, the visual representation of a conjunct glyph can be significantly different than the constituent consonants. Devanagari also uses a large number of characters (34 commonly used consonants, 14 commonly used independent vowels, 14 corresponding vowel modifiers, a *halant* to join consonants into conjuncts, and 3 commonly used diacritic marks, leading to about 66 unique, commonly used Unicode characters), which adds to the complexity of the text input task.

Researchers have explored the effect of adding speech recognition technologies to keyboards. Traditionally, researchers have reported text entry speeds ranging from 19 to 53 words per minute (WPM) (or 95 to 256 CPM) for English [2, 16, 17, 21]. After adding speech, Ruan et al. reported 161 WPM (about 805 CPM) for English, 108 WPM (about 540 CPM) for Mandarin Chinese as compared to 53 WPM (about 265 CPM) and 38 WPM (about 195 CPM) without speech, respectively [16, 17].

While speech may improve the speed, it may also cause more errors, and hence analysis of errors is important. Ruan et al. reported a mixed result. In their case, uncorrected error rates with speech were higher - 0.35% and 1.69% for English and Chinese respectively in contrast to 0.19% and 1.40% without speech. On the other hand, the corrected error rates with speech were lower at 2.58% and 5.8% compared to 3.49% and 19.14% without speech [16, 17]. Moreover, speech does not necessarily increase speed in all kinds of input tasks. For example, Rudnicky et al. [23] conducted a longitudinal study in which participants carried out 40 spreadsheet tasks alternating between keyboard and speech input. They observed that tasks took longer to finish through voice input.

As mentioned above, the much slower text entry rates have been reported for Indian languages, and speech seems to help particularly in these languages. After a between-subject longitudinal study with novice users lasting several weeks and providing about 300 minutes of controlled typing practice, we reported peak speeds between 35 to 45 CPM on four Marathi keyboards, namely Swarachakra Marathi,

CDAC InScript Devanagari, Swiftkey Marathi and Sparsh Marathi [8]. In our more recent study, we reported substantial gains by adding speech input to the Swarachakra Hindi keyboard [8]. We found that novice users could achieve mean speeds of up to 118 CPM with speech, compared to 47 CPM without speech. We found that speech increased both the uncorrected error rate (0.75% to 1.63%) and the corrected error rate (7.5% to 21.6%). The increase in the corrected error rate (unlike [16, 17]) implies that in the speech condition, users found and needed to correct many errors, which could be attributed to the relative immaturity of speech recognition technology for Indian languages.

While the smooth-screened smartphones are generally considered to be more "advanced" than the feature phones with hardware buttons, these were considered to be a "giant leap backwards" by the visually impaired users – especially for tasks such as text input. Fortunately, advances in accessibility research has led to somewhat more accessible text input methods. For example, Perkinput used Input Finger Detection (IFD) for non-visual touch screen input [3], which could then enable the visually impaired to input text with one hand. In their studies, users could achieve average text input speeds of 6 WPM (30 CPM) for Perkinput and 4 WPM (20 CPM) for VoiceOver. Their uncorrected error rates were also observed to be low for Perkinput at 3.52% and 6.43% for VoiceOver. Consequently, the corrected error rates were higher for Perkinput with an error rate of 12.23% and 8.32% for VoiceOver. Gaines modelled Tap123 after a standard QWERTY keyboard that does not require users to tap specific keys and achieved entry speeds of 19 WPM (95 CPM) and uncorrected error rates of 2.08% [2].

Much research has also been done using gestural interactions in accessible text entry. Kane et al. in their study described Slide Rule which is a gesture based technique that was compared with button based Pocket PC Screen Reader [4]. Although Slide Rule was significantly faster than the button-based system, more errors were found while using the gesture based system. On the contrary, NavTap used a navigational method and evaluated text entry speeds in real-life settings. The text entry speeds increased from 0.2-2.7WPM in the first session to 1.6-8.46 WPM in the 13th session over a period of 16 weeks. [9]

The advancement of text input research in Indian languages for visually impaired is limited. To the best of our knowledge, the only work in this area was conducted by Bharath et al., who conducted an empirical study that provided benchmark speeds for visually impaired users in Indic scripts [7]. They conducted a within subject study with two accessible keyboards – Google Indic and Swarachakra. The overall mean speeds were 14.2 CPM for Swarachakra and 12.8 CPM for Google Indic. The mean accuracy for Swarachakra was higher at 96% in contrast to 94% for Google Indic.

Only limited amount of prior work has been done for visually impaired people that analysed speech input in English. Azenkot et al. carried out a survey with 169 blind

and sighted users and later conducted a study with 8 blind users [1]. Their study evaluated the use of speech input on iPod vs on an on-screen keyboard. They found that although speech was 5 times faster than the keyboard, users spent an average of 80.3% of their time editing the errors. Bonner et al. describe No-Look Notes as an eyes-free text entry system that uses multi-touch input and audio output [22]. They evaluated No-Look Notes against Apple's accessibility component VoiceOver and found that No-Look Notes performed better than VoiceOver in terms of speed, accuracy and user preference. They found the overall speed for No-Look Notes to be 1.32 WPM (about 7 CPM) in contrast with 0.66 WPM (about 3 CPM) for VoiceOver.

To the best of our knowledge, there is no work reported that systematically evaluates the effect of applying speech recognition technologies to a given keyboard on the performance of visually impaired users. Further, we believe that we are the first to explore and evaluate speech as a method of text input in Indian languages for visually impaired people.

3 Keyboard Description

We conducted our study with the Swarachakra keyboard [5, 6, 7, 10] as it had emerged as the better performing keyboard in earlier studies for Marathi and Hindi. Swarachakra is a logically organised keyboard. The layout of the consonants in Swarachakra mimics the structure of the Devanagari script [6]. In the version for sighted users, when the user touches a consonant, Swarachakra displays a pie menu pop-up around the finger, which includes the 11 most frequently used vowel modifiers. The independent vowels are in a separate pie menu of its own. Swarachakra also supports previews of conjuncts, which is helpful for the sighted users.

Bharath et al adapted the design of Swarachakra to make it accessible [7]. In their variation, the interaction technique of the pie menu was changed. The user first explores the keyboard by touch to locate the desired consonant. As the user moves the finger, the screen reader reads out the consonant below the finger. Once the user reaches the desired consonant, she puts down a second finger, below which the vowel modifier pie menu is displayed (figure 2). The user can further explore the pie menu with the second finger until the desired vowel modifier is found. The keyboard has special gestures for backspace and for entering space.

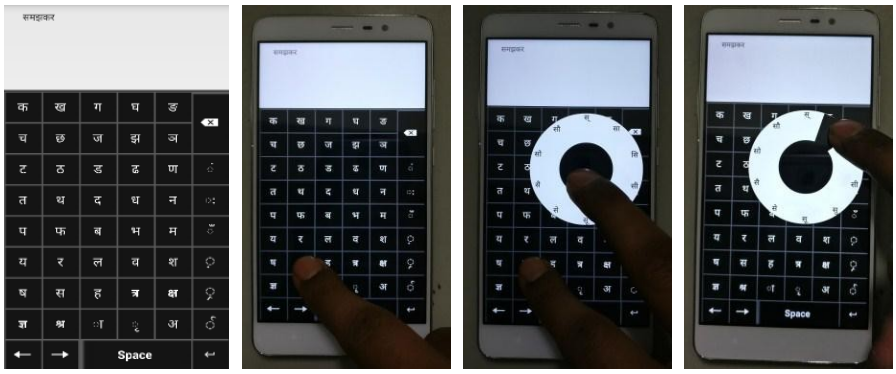

Fig. Error! No sequence specified.**.** Accessible version of Swarachakra from [7]. The user first explores by touch to locate the desired consonant while following the feedback from the screen reader. Once the desired consonant is located, she puts the second finger down under which a pie menu with 11 frequently used vowel modifiers is shown. The user further explores the pie menu by touch to select the desired vowel modifier.

Fig. Error! No sequence specified.**.** The modified layout of accessible Swarachakra layout that we used in our study. The interaction technique was identical to the one shown in figure 2.

Based on the user feedback from the study done by Bharath et al, and some pilot studies that we conducted, we modified the layout of the accessible version of Swarachakra slightly (figure 3). In the original design, the independent vowel pie menu was in the bottom row, and was difficult to locate. We moved it to the top row because independent vowels are reasonably frequent, and it is common for users to start exploring the keyboard from the top. In the original design, the less frequent vowel modifiers and diacritics such as rukar, anusvar, chandrabindu and visarga were absent. We added an additional pie menu of these characters in the top row alongside the independent vowel key. We moved other infrequent keys such as trakar, rafar and nukta to the rightmost column in their order of frequency. We added the navigation keys and the punctuation keys to the penultimate row and the Shift, Space and the Enter key to the last row of the layout.

In addition, we integrated Marathi speech recognition ability through the Liv.ai API[1]. When the user wishes to invoke speech recognition, she explores the keyboard till she reaches the microphone button in the third-last row on the right. When she reaches the button, she puts down a second finger anywhere on the keyboard. This invokes the speech recognition engine. When the speech engine is ready, it plays a beep sound, after which the user can speak into the phone. The user indicates that she has finished speaking by lifting the second finger. The keyboard plays another beep which indicates the end of listening and then relays the user's speech to the Liv.ai server. The server interprets the speech and sends back the recognised text, which the keyboard enters in the text box and also reads out through the screen reader. The user may then edit the text if she finds any recognition errors.

4 Method

The study protocol was a within-subject design that compared the performance of users in the keyboard-only condition with their performance in the keyboard+speech input condition. The protocol was partly derived from [5], [6], [7] and [8]. Each user did five tasks: a training task, a first-time usability task, a keyboard familiarization task, and two main tasks. The keyboard familiarization task and the two main tasks were longitudinal tasks – i.e. they were performed in multiple sessions spread across several days.

On the first day of the study, moderators trained the user to type and edit the texts in Marathi using the Accessible Swarachakra keyboard. The users were also familiarized with special gestures for backspace and for entering space. The users were trained on 10 words. All the tasks were conducted on a mobile application that displayed the words/phrases to be transcribed, and logged all user input, input time

[1] https://liv.ai/

and input errors. The application also gave audio feedback with regards to the accuracy of the phrase typed after the user submitted the phrase. The users were trained to read (with the screen reader) the phrase to be typed, the transcribed phrase, and the feedback.

After the training task, the user conducted a first-time usability task (FTU) by typing 20 words of various levels of difficulty. In the FTU, the user was allowed two attempts per word. In the first attempt no help was provided. If the user failed in the first attempt, a second attempt was allowed. Minimal help was provided in the second attempt if the users failed to type the word again. The user was considered to be trained successfully if she could type a significant proportion of words in the FTU without help.

After completion of the training and FTU on the first day, the users commenced the keyboard familiarization task. This task comprised of 15 sessions in each of which the user transcribed 8 phrases without any help from the moderator. If the user struggled to type a specific word during the session, the moderator provided additional practice for that word at the end of the session. We limited the number of such sessions that user could do in a single day to three, with an interval of at least 15 minutes between sessions. The training, the FTU task and the keyboard familiarisation task constituted the practice phase of the study.

At the conclusion of the practice phase, the users were assigned an input method for the first main task (keyboard-only or keyboard+speech input). Each main task comprised of 6 sessions in each of which the user transcribed 8 phrases. After completing the first main task, the users completed the second main task using the second input method. The sequence of input methods was counterbalanced across users. While performing the main tasks, we restricted the users to a maximum of two sessions in a day with at least 15 minutes between sessions.

Just before the users performed the main task involving the keyboard+speech input method, we gave a demonstration to the users about how they could use the speech input option. To locate the microphone button, the users were trained to use the right edge using their second digit and then to use the index finger to activate the speech service. To use the speech service, the users were asked to speak after they hear a beep and to lift both fingers after they finish speaking. The system reads back the recognized sentence after a beep. In case of an error, the participant could edit it using either the speech input or the keyboard.

To ensure that the users performed the tasks accurately and attain a good speed, we incentivized the users like Dalvi et al [6]. The users won a “speed prize” every time they typed a phrase at a speed higher than their previous best speed on a phrase. Besides the speed prizes, the users could also win an “accuracy prize” at the end of each session if they typed all the 8 phrases in that session with 100% accuracy. We describe how we calculated accuracy in the results section below.

4.1 Users and Study Context

We conducted the study with visually disabled children in a residential Marathi-medium school for the visually disabled girls in the city of Nashik, Maharashtra, India. Permissions were obtained from the school authorities and hostel authorities for the participation of the children in the study. The school in turn informed the parents of the children and sent them a copy of the project information. We recruited fourteen volunteers from classes seven to nine (all girls). All users were native Marathi speakers and were learning Marathi as their first language in the school. None of the users had used a smartphone or a computer previously. They could read and write the Braille Marathi script and used it for their academic work.

Fourteen users volunteered for the study. Out of these, 11 users completed all the sessions in time. Three users could not complete the study or had to take long breaks lasting several days between sessions. While we let them complete the study to the extent possible, to avoid bias, for the purpose of the analysis, we dropped these users from the results of this paper.

The sessions were carried out in an "office quiet" environment in a school classroom. At any given time, no more than 4 users performed the tasks in the room and they were supervised by at least 2 moderators. Prizes mentioned above were provided after consulting the school, and these included small items of stationery or cosmetics such as set of markers, drawing books, hair clips, hair ties, hair pins, pencil-box etc. At the end of the study, all users were given participation certificates and a "participation prize".

The study was conducted on Motorola G6 Android phones. We used a 4G cellular network to connect to the internet for speech recognition during the study.

4.2 Phrase set

We selected ten words for the training session and twenty words for the first-time usability test. Like [6], [7] and [8], these words were selected such that the users had enough opportunity to learn and explore typing in Marathi. For the keyboard familiarisation task, we selected 120 conversational phrases that were representative of everyday Marathi language used among native speakers. Phrases included conversational phrases, proverbs, lines from popular songs and poems, and phrases from school textbooks. Table 1 shows some examples of the selected phrases. For the main tasks, we selected another set of 48 similar phrases. The same sets of 48 phrases were used for both main tasks.

Table Error! No sequence specified.**.** Examples of selected phrases.

चहा गरम आहे	तो सकाळी लवकर उठला
तू कशी आहेस	सागरा प्राण तळमळला
दीपक पाणी आण	झाली सकाळ सरली रात
विजय पाट उचल	थेंबे थेंबे तळे साचे
किती वेळ लागेल	जजकडे वतकडे लख लख लख
आज खूप उकडत आहे	शरदने कॅमेऱ्याने फोटो काढले
अजयने चेंडू आणला	उषःकाल होताहोता काळरात्र झाली

5 Results

The main purpose of the study was to compare the user performance in the two conditions (keyboard-only and keyboard+speech input). During each of the two main tasks of the study, each user typed (8 sessions x 6 phrases =) 48 phrases that are relevant to our analysis. We first present the analysis of errors followed by typing speed of these phrases and then discuss the efficacy of text entry with and without speech.

Transcribing text with the help of a screen reader has several limitations for a visually impaired user, which leads to a peculiar set of errors in Indian languages that the user cannot avoid. These are similar to the problems faced by visually impaired users of English, where the user may occasionally miss an unwanted space (e.g. it is difficult to differentiate between “output” and “out put” with a screen reader). In Indian languages, it is particularly difficult to distinguishing between a long and a short vowel (e.g. the difference between the “u” sound of word “put” and the “oo” sound of the word “cool”). In our screen reader, we tried to enhance the difference by adjusting the tone of voice, but this too was not enough.

Hence, during the study, we used a “lenient” model for error calculation for vowel modifiers. Thus, while giving error feedback to the users and while calculating their eligibility for prizes, we tolerated errors such as substitution of a similar sounding vowel modifier, or an additional space. However, for the purpose of error analysis in this section of the paper, we report all the errors strictly. We computed uncorrected error rate (UER) and corrected error rate (CER) as described by Soukoreff et al [18].

Figure 4 below shows the CER and UER for the keyboard-only and keyboard+speech input conditions for all the sessions of the main tasks.

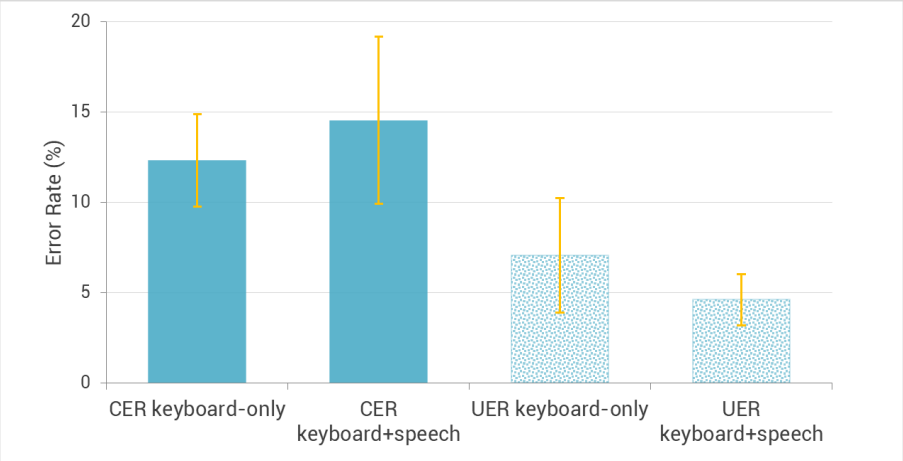

Fig. Error! No sequence specified.**.** Mean corrected error rate (CER) and uncorrected error rate (UER) for keyboard-only and keyboard+speech input modes. The error bars show 95% confidence intervals.

The mean UER for the keyboard-only condition was higher at 7.07% (N=11, SD = 4.76%, 95% CI 3.91% to 10.24%) than the UER for the keyboard+speech input condition, which was 4.61% (N=11, SD = 2.16%, 95% CI 3.18% to 6.04%). A paired t test revealed that the differences are not significant (N = 11, $p = 0.06$). Nevertheless, the direction of the difference is surprising, and contrary to results found in studies with sighted users in English, Chinese and Hindi ([16, 17, and 8] respectively), where UER was found to be higher for speech. This suggests that visually impaired users in our study did not notice some errors that were more evident to sighted users, and hence left them uncorrected.

The mean CER for the keyboard-only condition was lower at 12.33% (N=11, SD = 3.85%, 95% CI 9.78% to 14.89%) than keyboard+speech input condition, which was 14.53% (N=11, SD = 6.98%, 95% CI 9.90% to 19.17%). A paired t test reveals that the differences are not significant (N = 11, $p = 0.44$). The direction of this result is consistent with the results found in [8], a study with sighted users in Hindi, though inconsistent with [16, 17], a study with sighted users in English and Chinese. This is probably due to the speech recognition accuracy of Indian languages.

Figures 5 and 6 show session-wise corrected and uncorrected error rates for keyboard-only method and keyboard+speech method. In both the input methods, the

UER had little variation across sessions, while CER tended to fall as is visible from the trendlines in the graphs. We can attribute this to a practice effect in both conditions, and speculate that CER could reduce further with more practice.

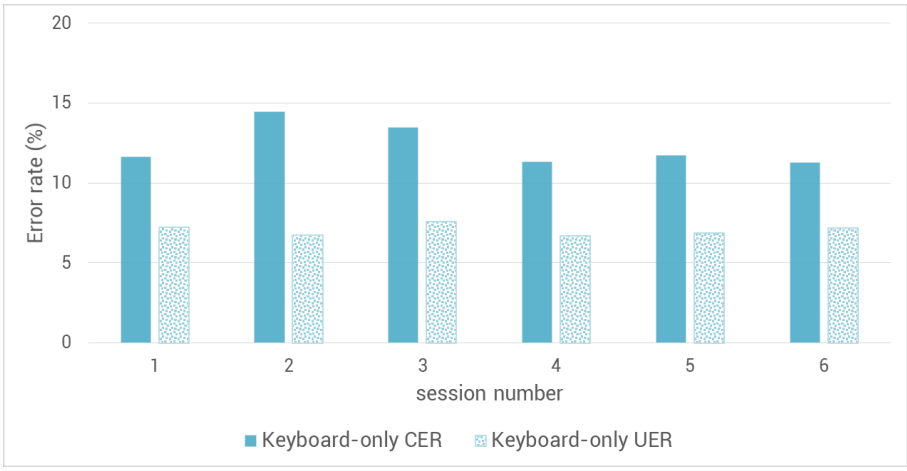

Fig. Error! No sequence specified.**.** Session-wise mean corrected and uncorrected error rates for keyboard-only input mode.

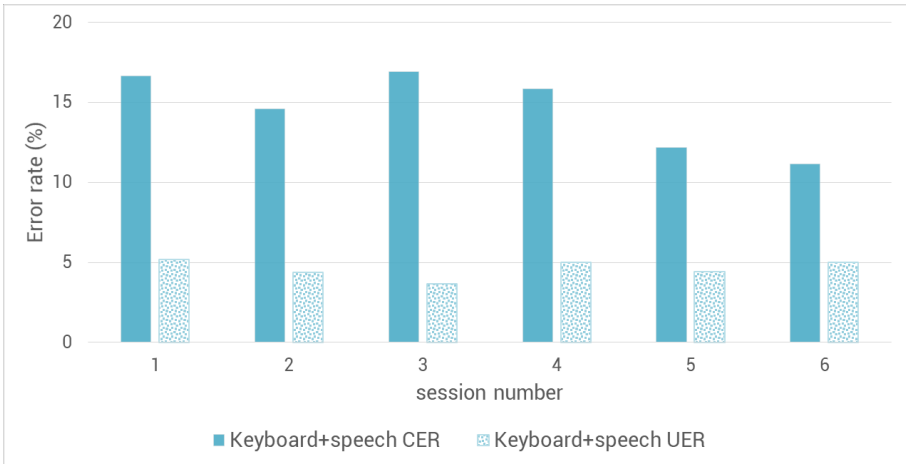

Fig. Error! No sequence specified.**.** Session-wise mean corrected and uncorrected error rates for keyboard+speech input mode.

We compared the accuracy of the typed phrases in both the input methods. We found that out of 48 phrases, only three phrases (6%) had an accuracy of 100% in the keyboard+speech input method for all 11 users. The lengths of such phrases were between 12-25 characters and the average time required was 6.3533 seconds. These phrases are किती वेळ लागेल, दुष्काळात तेरावा महिना and तो सकाळी लवकर उठला. Two of these phrases are conversational phrases and one is a proverb. The average accuracy for these phrases using the Keyboard-only method was 94.48% and time was 70.23%.

To calculate speed, we used a similar method as reported by Bhikne et al in their study [8]. For the keyboard-only method, the phrase task time was considered from the time the user typed the first character till the time of the user made the last alteration to the transcribed phrase. If n was the number of Unicode characters in the typed phrase, the speed was calculated by dividing n-1 by the phrase task time in minutes. For the keyboard+speech input method, the phrase task time was considered from the time the user pressed the mic button till the time the user made the last modification to the typed phrase. In contrast to the keyboard-only method, the speed for the keyboard+speech input method was calculated by dividing n by the phrase task time, where n is the number of Unicode characters typed by the user.

Most often, users submitted well-formed phrases with an occasional uncorrected error. On rare occasions though, users accidentally pressed the submit button of the logging tool before they meant to, perhaps while they were exploring the keyboard by touch. In such cases, this led to a unusually high uncorrected error rate for that phrase, and often, an unusually high input speed. We attribute this higher speed to the study situation rather than to the input method. To reduce this bias, we dropped phrases that had an uncorrected error rate of more than 20% for the purpose of analysis of speed. Out of the total of (2 tasks x 6 sessions x 8 phrases per session x 11 users =) 1,056 phrases that were typed during the main tasks, 66 phrases (6.25%) had an uncorrected error rate of more than 20%, and were dropped in this way. We note here that this number is much higher than the study reported with sighted users [8] who had reported only 0.75% such phrases.

Figure 7 shows the results of the analysis of speed differences. The keyboard-only condition had a mean speed of 16.04 CPM (N=11, SD = 5.22, 95% CI 12.58 to 19.50). The keyboard+speech input condition had a mean speed of 182.13 CPM (N=11, SD = 48.05, 95% CI 150.25 to 214.02). As can be guessed, a paired t test revealed that the differences are significant ($N = 11$, $p < 0.0005$).

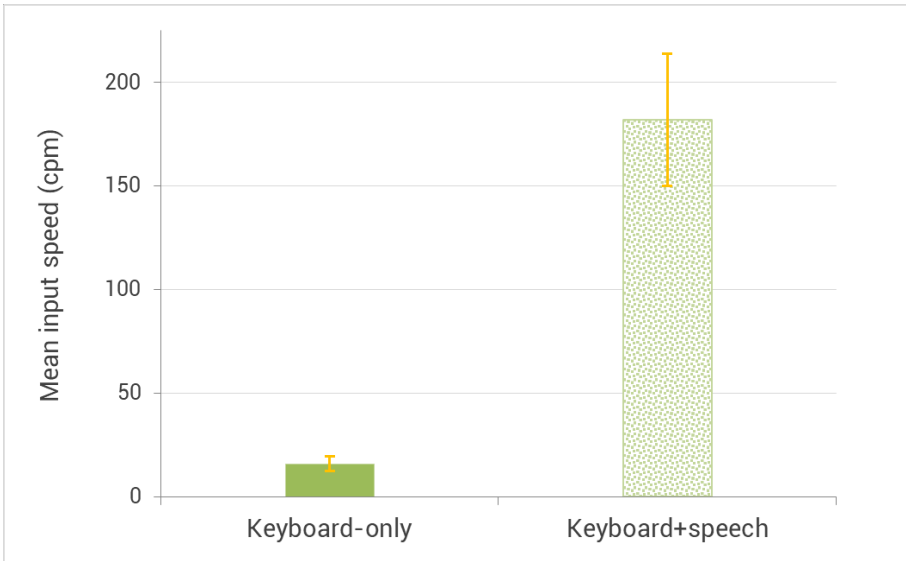

Fig. Error! No sequence specified.**.** Mean input speed (CPM) for keyboard-only and keyboard+speech input modes. The error bars show 95% confidence intervals.

To investigate the effect of speech recognition engine on the performance of the user, we had selected a phrase set consisting of a variety of phrases including popular poems, songs, proverbs and conversational phrases. Similar to Bhikne et al. [8], we created a "ground truth" of the phrase recognition accuracy in our lab. Two expert users who were native Marathi speakers spoke out the phrases in an "office-quiet" environment. A phrase was determined to be "completely recognised" (CR) and requiring no edits if the transcribed phrase matched with the given phrase for both experts. If either of the experts had any phrases that needed editing, the phrase was determined to be "partially recognised" (PR). Of the 48 phrases, 29 phrases (60.41%) were completely recognised, while 19 phrases (39.58%) were partially recognised. There were no phrases that were not recognised by the speech recognition engine.

We compared the "ground-truth" with the performance of the users. As expected, the results varied somewhat from the ground truth. Out of the 528 phrases that were typed using the keyboard+speech input by the 11 users, 275 (51.98%) phrases were completely recognised (CR). Another 248 of the 528 (46.88%) phrases were partially recognised with an accuracy of more than 65%. Only 6 phrases (1.13%) were not recognised at all when the users spoke them, or had an accuracy of less than 65%.

We also performed a phrase-wise analysis to observe the "underwater" phrases. As defined in [8], phrases that were slower in the keyboard+speech input method than the keyboard-only method are said to be "underwater phrases" while the other phrases are

said to be “above water phrases”. Of the 48 phrases that were typed, on an average, all phrases were faster with keyboard+speech input method than keyboard-only method by more than 10%. This contrasts with the findings from [8], a study done with sighted users in Hindi, where 12.7% phrases were found to be “underwater”. This could be partly attributed to the difference in language (Hindi vs. Marathi), but much more substantially to the fact that visually impaired users had a much lower base rate for text input in keyboard-only method than sighted users.

To explore if there is any correlation between the mean speeds of the users in the keyboard-only method and keyboard+speech method, we calculated Pearson’s Correlation Moment. There was a low negative correlation between the speeds of the users in the two input methods but the correlation is not significant (r = -0.347, p = 0.295). Figure 7 illustrates the scatter plot for the two distributions. It is possible that users who type faster with the keyboard-only method perform somewhat slower with the keyboard+speech input method than users who are not so fast. The other possibility is that the skills required to type quickly with the two methods are independent of each other. Please note that our study is quite small (N=11), and it needs to be repeated with a larger sample for strengthening either of these results.

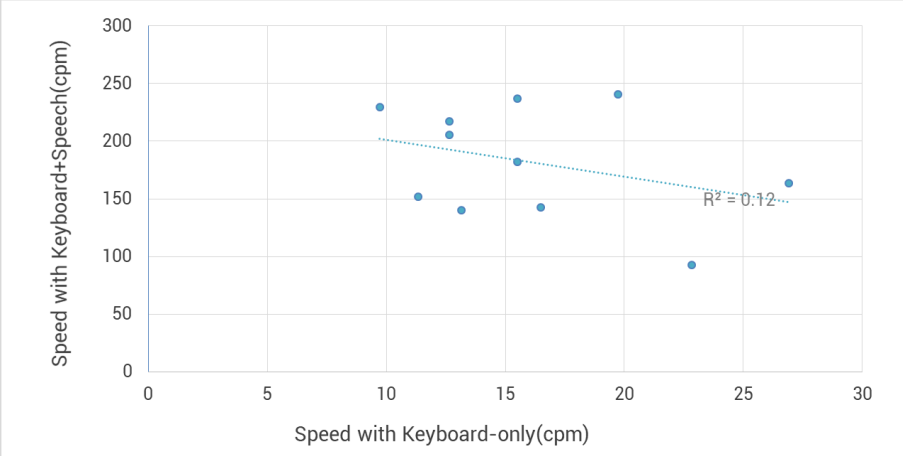

Fig. Error! No sequence specified.**.** Pearson's Correlation moment between the mean speeds of each user (n=11, r = -0.347, p = 0.295).

5.1 Other findings

There were some other interesting findings during our study. The layout of the Swarachakra keyboard is based on the sequence and the structure of the Devanagari

script. The sequence (and the structure) is taught to sighted children from childhood. Novice users use this sequence / structure to locate keys on the keyboard. This has been one of the strengths of the design of the Swarachakra keyboard. However, during our study we learnt that Devanagari Braille is taught in a different sequence and thus the visually impaired children are not familiar with the original sequence / structure of Devanagari. To an extent, this hampered the learning of the keyboard. While this issue is relatively less important in this paper, which aims to compare the performance of visually impaired users with speech enabled keyboards, it could have a broader implication on the design of keyboards.

We tried to analyse the speech recognition errors, and found some interesting patterns. As could be expected, some words were misrecognised as similar sounding more frequent words or spelling alternatives (e.g. पॅट as पॅन्ट, मुलं as मुले). We found that words with lower frequencies such as झरझर and कुऱ्हाडीचा were misrecognized as जर्जर (which is completely different, though similar sounding word) and कुर्हाडीचा (which is a wrong spelling). More popular words including words such as दुसऱ्या, कॅमेऱ्याने and सरड्याची, though arguably equally complex to type, were recognized accurately.

When a phrase contained clusters of repeated words for a poetic effect (e.g. झुक झुक झुक झुक and लख लख लख) such clusters were often recognized as a single words, i.e. without the spaces (e.g. झुकझुकझुकझुक and लखलखलख respectively). This could be a result of the users speaking the phrases without pauses or in a rhythm.

6 Conclusion

We conducted an empirical study with 11 visually impaired children to compare performances of keyboard-only method with keyboard+speech input method. We observed that in spite of speech recognition errors, the keyboard+speech input method was almost 11 times faster than keyboard-only method.

Ours is the first empirical study that evaluates the two input modalities of text entry in Indian languages. We also document the highest ever text entry speeds reported for Indian languages by visually impaired users, and, in fact, by any group of users. It is interesting to note that the speeds achieved by the visually impaired users in our study with keyboard+speech input was 182 CPM, which was substantially above the speed reported by Bhikne et al [8] of 118 CPM by sighted users. The languages used in the two studies were different (Marathi and Hindi). Consequently, phrase sets were different. Also, the user groups were different. Hence the results of the two studies are not strictly comparable. Nevertheless, the two languages are related enough, and the phrase sets, the methods and users in the two studies are similar enough to interest us in a future study to compare such effects more systematically.

Of the 48 phrases that were typed in our study, none were “underwater”. That means, on an average, all phrases were faster with keyboard+speech input method than the keyboard-only method by more than 10%. This contrasts with the findings from [8], a study done with sighted users in Hindi, where 12.7% phrases were found to be “underwater”. This is possibly because the baseline speeds of typing (i.e. in the keyboard-only mode) is higher for sighted users than the visually impaired users. In other words, speech input helps the visually impaired users a lot more than it helps the sighted users. Of course, as noted above, the studies are not strictly comparable, as the languages of the two studies are different. Future work could explore this finding more systematically.

We were surprised to find that the uncorrected error rate (UER) for the keyboard-only condition was higher than for keyboard+speech input condition, though the difference is not significant. This is contrary to reported studies with sighted users in English, Chinese and Hindi [16, 17, 8], where UERs were found to be higher for speech. UER for sighted users in keyboard-only condition was reported at only 0.75% for Hindi, 1.40% for Mandarin and 0.19% for English, compared to 7.07% in our study with visually impaired users in Marathi.

One possible explanation for this could be that at the baseline (i.e. without speech) visually impaired users leave behind a larger number of uncorrected errors in the text because they are not able to detect such errors with the help of screen readers. As mentioned above, short and long vowel errors in Indian languages are particularly hard to differentiate with screen readers. When automatic speech recognition engines recognize text, such systems use dictionaries, which do not have not have too many errors of those kinds. On the other hand, speech recognition systems create errors of their own (which could be of a different nature than the ones that creep in because of screen readers), and a sighted person may miss correcting those errors, as his mind is already conditioned by the “correct speech” he believes he has said, and then such errors get left behind uncorrected in the transcribed text, leading to higher UER in the speech condition for sighted users. A visually impaired user may somehow have been alert to such errors, again perhaps because of the screen reader interface. Future research could analyse the different types of errors made by users and the role of error perception in different media (visual vs. audio).

Consistent with [8] (and in turn, inconsistent with [16 and 17]), we found that corrected error rates for the keyboard+speech condition were higher than the keyboard-only condition (though our differences are not significant). This implies that our users (and those in [8]) found and corrected many errors in speech recognition, implying that the speech recognition technologies for Indian languages have yet to mature.

In both the input methods, the UER had little variation across sessions, while CER tended to fall with sessions. This is understandable as our study was done with novice

users. With practice, the need for error correction seems to be going down in both keyboard-only and keyboard+speech input conditions. Correspondingly, speeds seem to be still on the rise. Future work needs to investigate effects of even longer term practice, by perhaps working with expert users.

We acknowledge that with N=11, our study was quite small. Given that text input in Indian languages by visually impaired users on mobile phones is new, it was necessary for us to train up novice users and, this implied a longitudinal study. Given the constraints of resources and logistics, and availability of visually impaired users, this was the best that we could do in this study. Yet, we believe that we have contributed a lot to the knowledge in this space, and we hope that our study will light the way forward for future research in this area.

Acknowledgements

This project was funded by Tata Consultancy Services. We are thankful to all our participants and administration from National Association for Blind in Nashik, India for their invaluable feedback and support.

References

1. Shiri Azenkot and Nicole B. Lee. 2013. Exploring the use of speech input by blind people on mobile devices. In Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13). ACM, New York, NY, USA, , Article 11 , 8 pages. DOI: http://dx.doi.org/10.1145/2513383.2513440
2. Dylan Gaines. 2018. Exploring an Ambiguous Technique for Eyes-Free Mobile Text Entry. In Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '18). ACM, New York, NY, USA, 471-473. DOI: https://doi.org/10.1145/3234695.3240991
3. Shiri Azenkot, Jacob O. Wobbrock, Sanjana Prasain, and Richard E. Ladner. 2012. Input finger detection for nonvisual touch screen text entry in Perkinput. In Proceedings of Graphics Interface 2012 (GI '12). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 121-129.
4. Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (Assets '08). ACM, New York, NY, USA, 73-80. DOI: https://doi.org/10.1145/1414471.1414487
5. Girish Dalvi, Shashank Ahire, Nagraj Emmadi, Manjiri Joshi, Nirav Malsettar, Debasis Samanta, Devendra Jalihal, and Anirudha Joshi. 2015. A Protocol to Evaluate Virtual Keyboards for Indian Languages. In Proceedings of the 7th International Conference on

HCI, IndiaHCI 2015 (IndiaHCI 2015). ACM, New York, NY, USA, 27-38. DOI: https://doi.org/10.1145/2835966.2835970

6. Girish Dalvi, Shashank Ahire, Nagraj Emmadi, Manjiri Joshi, Anirudha Joshi, Sanjay Ghosh, Prasad Ghone, and Narendra Parmar. 2016. Does prediction really help in Marathi text input?: empirical analysis of a longitudinal study. In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2016). ACM, New York, NY, USA, 35-46. DOI: https://doi.org/10.1145/2935334.2935366
7. Anu Bharath P., Jadhav C., Ahire S., Joshi M., Ahirwar R., Joshi A. (2017) Performance of Accessible Gesture-Based Indic Keyboard. In: Bernhaupt R., Dalvi G., Joshi A., K. Balkrishan D., O'Neill J., Winckler M. (eds) Human-Computer Interaction - INTERACT 2017. INTERACT 2017. Lecture Notes in Computer Science, vol 10513. Springer, Cham
8. Bhakti Bhikne, Anirudha Joshi, Manjiri Joshi, Shashank Ahire, and Nimish Maravi. 2018. How Much Faster Can You Type by Speaking in Hindi?: Comparing Keyboard-Only and Keyboard+Speech Text Entry. In Proceedings of the 9th Indian Conference on Human Computer Interaction (IndiaHCI 2018). ACM, New York, NY, USA, 20-28. DOI: https://doi.org/10.1145/3297121.3297123
9. Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonsalves. 2009. NavTap: a long term study with excluded blind users. In Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (Assets '09). ACM, New York, NY, USA, 99-106. DOI: https://doi.org/10.1145/1639642.1639661
10. Anirudha Joshi, Girish Dalvi, Manjiri Joshi, Prasad Rashinkar, and Aniket Sarangdhar. 2011. Design and evaluation of Devanagari virtual keyboards for touch screen mobile phones. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2011). ACM, New York, NY, USA, 323-332. DOI: https://doi.org/10.1145/2037373.2037422
11. Google TalkBack. Wikipedia. Retrieved January 27, 2019 from https://en.wikipedia.org/w/index.php?title=Google_TalkBack&oldid=849832493
12. VoiceOver. Wikipedia. Retrieved January 27, 2019 from https://en.wikipedia.org/w/index.php?title=VoiceOver&oldid=870848560
13. Manoj Kumar Sharma and Debasis Samanta. 2014. Word Prediction System for Text Entry in Hindi. 13, 2, Article 8 (June 2014), 29 pages. DOI: https://doi.org/10.1145/2617590
14. Younghee Jung, Dhaval Joshi, Vijay Narayanan-Saroja, and Deepak Prabhu Desai. 2011. Solving the great Indian text input puzzle: touch screen-based mobile text input design. In proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2011). ACM, New York, NY, USA, 313-322. DOI: https://doi.org/10.1145/2037373.2037421
15. Lauren Hinkle, Albert Brouillette, Sujay Jayakar, Leigh Gathings, Miguel Lezcano, and Jugal Kalita. 2013. Design and Evaluation of Soft Keyboards for Brahmic Scripts. 12, 2, Article 6 (June 2013), 37 pages. DOI=http://dx.doi.org/10.1145/2461316.2461318
16. Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Y. Ng, and James A. Landay. 2016. Speech Is 3x Faster than Typing for English and Mandarin Text Entry on Mobile Devices. CoRR abs/1608.07323.

17. Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1, 4: 1–23. https://doi.org/10.1145/3161187
18. R. William Soukoreff and I. Scott MacKenzie. 2004. Recent developments in text-entry error rate measurement. In CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04). ACM, New York, NY, USA, 1425-1428. DOI: https://doi.org/10.1145/985921.986081
19. List of languages by number of native speakers in India - Wikipedia. Retrieved January 28, 2019 from https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India
20. Devanagari - Wikipedia. Retrieved January 28, 2019 from https://en.wikipedia.org/wiki/Devanagari
21. Scott I. MacKenzie, and William R. Soukoreff. 2002. Text entry for mobile computing: Models and methods, theory and practice. Human--Computer Interaction, 17, 2--3: 147--198.
22. Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, and W. Keith Edwards. 2010. No-Look Notes: Accessible Eyes-Free Multi-touch Text Entry. In Pervasive Computing, Patrik Floréen, Antonio Krüger and Mirjana Spasojevic (eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 409–426. https://doi.org/10.1007/978-3-642-12654-3_24
23. Alexander I. Rudnicky, Michelle Sakamoto, and Joseph H. Polifroni. 1989. Evaluating spoken language interaction. In Proceedings of the workshop on Speech and Natural Language (HLT '89). Association for Computational Linguistics, Stroudsburg, PA, USA, 150-159. DOI: https://doi.org/10.3115/1075434.1075459